Implementation Guide

Streaming Data Solution for Amazon MSK

Streaming Data Solution for Amazon MSK: Implementation Guide

Table of Contents

Solution overview

Publication date: *November 2020 (last update: June 2024)*

The Streaming Data Solution for Amazon MSK allows you to capture, store, process, and deliver real-time streaming data. By automatically configuring the included AWS services, this solution helps you address real-time streaming use cases, for example:

- Capture high volume application log files
- Analyze website clickstreams
- Process database event streams
- Track financial transactions
- Aggregate social media feeds
- Collect IT log files
- Continuously deliver to a data lake

This solution helps accelerate your development lifecycle by minimizing or eliminating the need to model and provision resources using AWS CloudFormation, set up preconfigured Amazon CloudWatch alarms set to recommended thresholds, dashboards, and logging, and manually implement streaming data best practices. This solution is data and logic agnostic, meaning that you can start with boilerplate code and then customize it to your needs.

The solution uses templates where data flows through producers, streaming storage, consumers, and destinations. Producers continuously generate data and send it to streaming storage where it is durably captured and made available for processing by a data consumer. Data consumers process the data and then send it to a destination.

To support multiple use cases and business needs, this solution offers four AWS CloudFormation templates. You can use this solution to test new service combinations as the basis for your production environment, and to improve existing applications.

1. **Option 1** creates a standalone Amazon Managed Streaming for Apache Kafka (Amazon MSK) cluster following best practices, such as sending broker logs to Amazon CloudWatch Logs; encryption at rest; encryption in transit among the broker nodes; and open monitoring with Prometheus activated.

2. **Option 2** adds an AWS Lambda function that processes records in an existing Apache Kafka topic as a starting example that you can modify and customize. The Lambda service internally polls for new records or messages from the event source, and then synchronously invokes the target Lambda function.
3. **Option 3** is intended for use cases when you must back up messages from a topic in Amazon MSK (for instance, to replay or analyze them). It uses Amazon Data Firehose (which compresses and encrypts, minimizing the amount of storage used at the destination and increasing security) and Amazon Simple Storage Service (Amazon S3).
4. **Option 4** showcases how to read data from an existing topic in Amazon MSK using Apache Flink, which provides exactly-once processing. It uses Amazon Managed Service for Apache Flink (a fully managed service that handles core capabilities like provisioning compute resources, parallel computation, automatic scaling, and application backups) and Amazon Simple Storage Service (Amazon S3).

All templates are configured to apply best practices to monitor functionality using dashboards and alarms, and to secure data.

This implementation guide describes architectural considerations and configuration steps for deploying the Streaming Data Solution for Amazon MSK in the Amazon Web Services (AWS) Cloud. It includes links to AWS CloudFormation templates that launch and configure the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for IT architects, developers, and DevOps professionals who want to get started quickly with the core streaming services available in the AWS Cloud.

This solution is a demo. We do not recommend using this to handle regulated data such as PII, HIPAA, and GPDR when deployed in production.

Features and benefits

The Streaming Data Solution for Amazon MSK provides the following features:

Automated configuration

Automatically configure the AWS services necessary to easily capture, store, process, and deliver streaming data.

Four template options

You can choose from four **AWS CloudFormation**template options. You can more quickly test new service combinations for your production environment and improve existing applications.

Real-time use cases

You can capture high-volume application logs, analyze clickstream data, continuously deliver to a data lake, and more.

Preconfigured Amazon CloudWatch alarms, dashboards, and logging

This solution comes with Amazon CloudWatch alarms set to recommended thresholds, dashboards for viewing performance metrics, and logging to make it easier for you to monitor the overall performance of the solution.

Customizable source code

Customize the solution's boilerplate code, and then use the monitoring capabilities to quickly transition from testing to production.

Integration with AWS Service Catalog AppRegistry and AWS Systems Manager Application Manager

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both AWS Service Catalog AppRegistry and AWS Systems Manager Application Manager. With this integration, you can centrally manage the solution's resources.

Use cases

Option 1: Standalone Amazon MSK cluster

This option creates a standalone Amazon Managed Streaming for Apache Kafka (Amazon MSK) cluster following best practices, such as sending broker logs to Amazon CloudWatch Logs; encryption at rest; encryption in transit among the broker nodes; and open monitoring with Prometheus activated.

Option 2: Add a Lambda function to process records

This option adds an AWS Lambda function that processes records in an existing Apache Kafka topic as a starting example that you can modify and customize. The Lambda service internally polls for new records or messages from the event source, and then synchronously invokes the target Lambda function.

Option 3: Backup messages from Amazon MSK to S3

This option is intended for use cases when you must back up messages from a topic in Amazon MSK (for instance, to replay or analyze them). It uses Amazon Kinesis Data Firehose (which compresses and encrypts, minimizing the amount of storage used at the destination and increasing security) and Amazon Simple Storage Service (Amazon S3).

Option 4: Analyze and store messages from Amazon MSK

This option showcases how to read data from an existing topic in Amazon MSK using Apache Flink, which provides exactly-once processing. It uses Amazon Kinesis Data Analytics(a fully managed service that handles core capabilities like provisioning compute resources, parallel computation, automatic scaling, and application backups) andAmazon Simple Storage Service (Amazon S3).

Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the monthly cost for running this solution in the US East (N. Virginia) Region, is described in the following tables.

Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution. We recommend creating a budget through AWS Cost Explorer to help manage costs.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

Sample cost tables

Option 1: Deploy the AWS CloudFormation template using Amazon Managed Streaming for Apache Kafka (Amazon MSK)

The following table provides a cost estimate to deploy the `streaming-data-solution-for-msk` AWS CloudFormation template that deploys Amazon MSK.

Table for Option 1: Cost estimate for running the solution using the CloudFormation template that deploys Amazon MSK

AWS service	Dimensions	Cost [USD]
Amazon MSK	Broker instance type: kafka.m5.large (3 nodes) Broker storage: 1,000 GB	$468.72 $100.00
Amazon EC2	EC2 instance (t3.small) 730 hours / month	$15.18
	TOTAL:	**$583.90 per month**

Note

The templates for options 2, 3 and 4 accept the Amazon Resource Name (ARN) of the Amazon MSK cluster as a parameter, so the following cost tables only include the services created by this solution.

Option 2: Deploy the AWS CloudFormation template using Amazon MSK and AWS Lambda

The Option 2 table provides a cost estimate to deploy the `streaming-data-solution-for-msk-using-aws-lambda` AWS CloudFormation template that uses Amazon MSKand Lambda.

Table for Option 2: Cost estimate for running the solution using the CloudFormation template that deploys Amazon MSK and Lambda

AWS service	Dimensions	Cost [USD]
AWS Lambda	2,678,400 requests/month (1/sec) 128 MB of memory 500 ms/request	$3.33
	TOTAL:	**$3.33 per month**

Option 3: Deploy the AWS CloudFormation template using Amazon MSK, AWS Lambda, and Amazon Data Firehose

The following table provides a cost estimate to deploy the `streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose` AWS CloudFormation template that uses Amazon MSK, AWS Lambda, Firehose, and Amazon Simple Storage Service (Amazon S3).

Table for Option 3: Cost estimate for running the solution using the AWS CloudFormation template that deploys Amazon MSK, Lambda, Firehose, and Amazon S3

AWS service	Dimensions	Cost [USD]
Lambda	2,678,400 requests/month (1/sec) 128 MB of memory 500 ms/request	$3.33
Firehose	100 records (4 KB)/second	$36.34
Amazon S3	1 GB storage (Amazon S3 standard)	$0.02
	TOTAL:	**$39.69 per month**

Option 4: Deploy the AWS CloudFormation template using Amazon MSK, Amazon Managed Service for Apache Flink, and Amazon S3

The following table provides a cost estimate to deploy the `streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3` AWS AWS CloudFormation template that uses Amazon MSK, Amazon Managed Service for Apache Flink, and Amazon Simple Storage Service (Amazon S3).

Table for Option 4: Cost estimate for running the solution using the AWS CloudFormation template that deploys Amazon MSK, Amazon Managed Service for Apache Flink, and Amazon S3

AWS service	Dimensions	Cost [USD]
Managed Service for Apache Flink	1 processing unit 50 GB running application storage	$80.30 $5.00
Amazon S3	1 GB storage (Amazon S3 standard)	$0.02
	TOTAL:	**$85.32 per month**

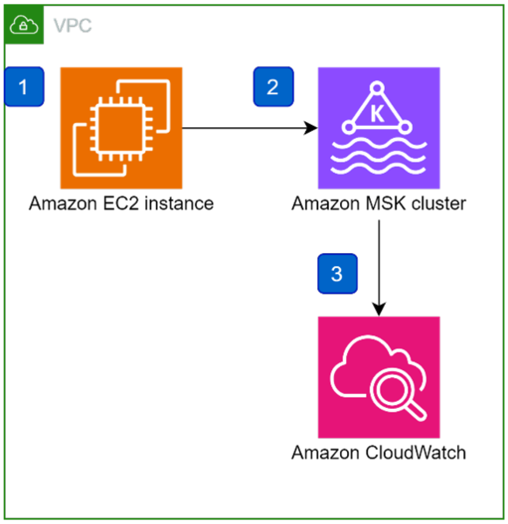
Architecture overview

This solution automatically configures the core AWS services necessary to capture, store, process, and deliver streaming data.

All AWS CloudFormation resources were created using AWS Solutions Constructs.

Option 1: Deploy the AWS CloudFormation template using Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Deploying the `streaming-data-solution-for-msk` AWS CloudFormation template builds the following environment in the AWS Cloud.



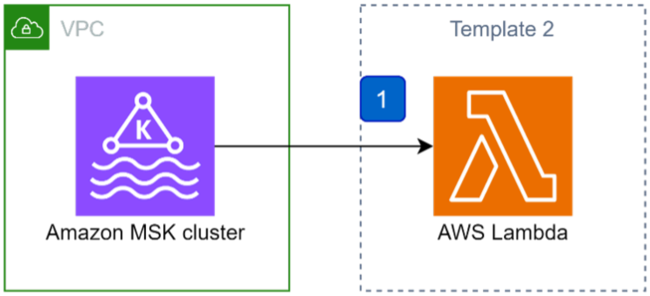
AWS CloudFormation template using Amazon MSK reference architecture

This AWS CloudFormation template deploys a reference architecture that includes the following:

1. An Amazon MSK cluster.
2. An Amazon EC2 instance that contains the Apache Kafka client libraries required to communicate with the MSK cluster. This client machine is located on the same VPC as the cluster, and it can be accessed via AWS Systems Manager Session Manager.
3. An Amazon CloudWatch dashboard monitors application health, progress, resource utilization, events, and errors.

Option 2: Deploy the AWS CloudFormation template using Amazon MSK and AWS Lambda

Deploying the `streaming-data-solution-for-msk-using-aws-lambda` AWS CloudFormation template builds the following environment in the AWS Cloud.



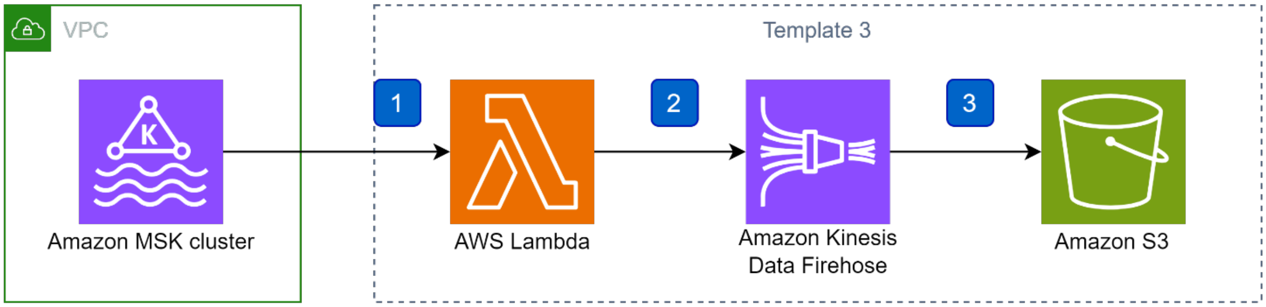
AWS CloudFormation template using Amazon MSK and Lambda reference architecture

This AWS CloudFormation template deploys a reference architecture that includes the following:

1. A Lambda function that processes process records in a Kafka topic. The default function is a Node.js application that logs the received messages, but it can be customized to fit your business needs.

Option 3: Deploy the AWS CloudFormation template using Amazon MSK, AWS Lambda, and Amazon Data Firehose

Deploying the `streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose` AWS CloudFormation template builds the following environment in the AWS Cloud.



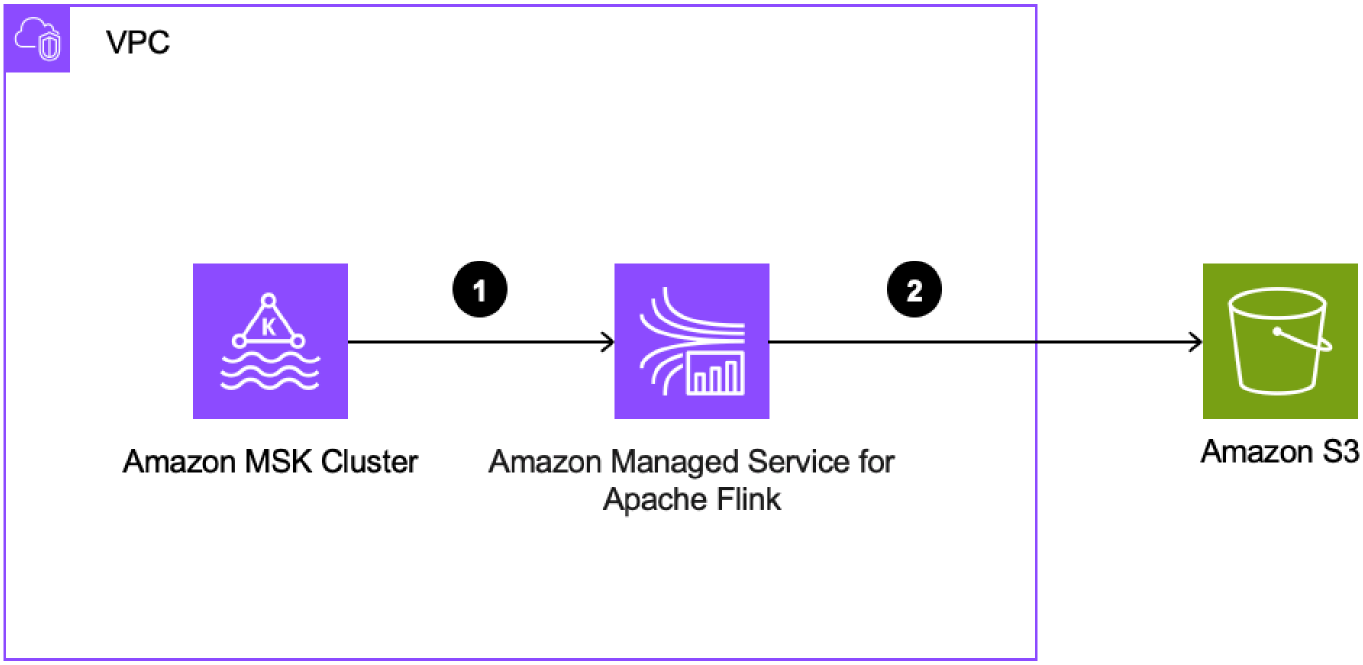
AWS CloudFormation template using Kinesis Data Streams, Kinesis Data Firehose, and S3 reference architecture

This AWS CloudFormation template deploys a reference architecture that does the following:

1. An AWS Lambda function that processes process records in an Apache Kafka topic.
2. A Firehose delivery stream that buffers data before delivering it to the destination.
3. An Amazon S3 bucket that stores all original events from the Amazon MSK cluster.

Option 4: Deploy the AWS CloudFormation template using Amazon MSK, Amazon Managed Service for Apache Flink, and Amazon S3

Deploying the `streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3` AWS CloudFormation template builds the following environment in the AWS Cloud.



AWS CloudFormation template using Amazon MSK, Amazon Managed Service for Apache Flink, and Amazon S3 reference architecture

This AWS CloudFormation template deploys a reference architecture that includes the following:

1. A Amazon Managed Service for Apache Flink Studio notebook application that reads events from an existing topic in an Amazon MSK cluster.
2. An AWS Glue Data Catalog to store metadata tables representing data stores.
3. An Amazon S3 bucket that stores the output.

AWS Well-Architected design considerations

This solution was designed with best practices from the AWS Well-Architected Framework which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework were applied when building this solution.

Ingesting and processing real-time streaming data requires scalability and low latency to support a variety of applications such as activity tracking, transaction order processing, click-stream analysis, data cleansing, metrics generation, log filtering, indexing, social media analysis, and IoT device data telemetry and metering. These applications are often spiky and process thousands of events per second.

Operational excellence

This section describes how we architected this solution using the principles and best practices of the operational excellence pillar.

The Streaming Data Solution for Amazon MSK solution pushes metrics to Amazon CloudWatch to provide observability into the infrastructure; AWS Lambda functions, Kinesis Data Analytics, Kinesis Data Firehose, S3 buckets, and the rest of the solution components.

Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

- All data storage including Amazon S3 buckets have encryption at rest.
- All inter-service communications use AWS IAM roles.
- Communications between end user and Amazon API Gateway uses Bearer token generated and handed by Amazon Cognito.

- All roles used by the solution follows least-privilege access. That is, it only contains minimum permissions required so the service can function properly.

Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

The Streaming Data Solution for Amazon MSK solution uses AWS Serverless services wherever possible (examples include AWS Lambda and Amazon S3) to ensure high availability and recovery from service failure.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

- Using serverless architecture throughout this solution.
- The ability to launch this solution in any region that supports AWS services in this solution such as: Amazon MSK, Kinesis Data Analytics, Kinesis Data Firehose, EC2, S3 Bucket, CloudWatch, and AWS Lambda.
- Multiple options are available to quickly carry out comparative testing using different types of service configurations.

Cost optimization

This section describes how we architected this solution using the principles and best practices of the cost optimization pillar.

- Using serverless architecture so that customers only get charged for what they use.
- Providing an option to the user on whether or not to enable enhanced monitoring (shard-level) for Amazon Kinesis Data Streams. This option is turned off by default to reduce the cost for users who don’t need shard-level data monitoring.

Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

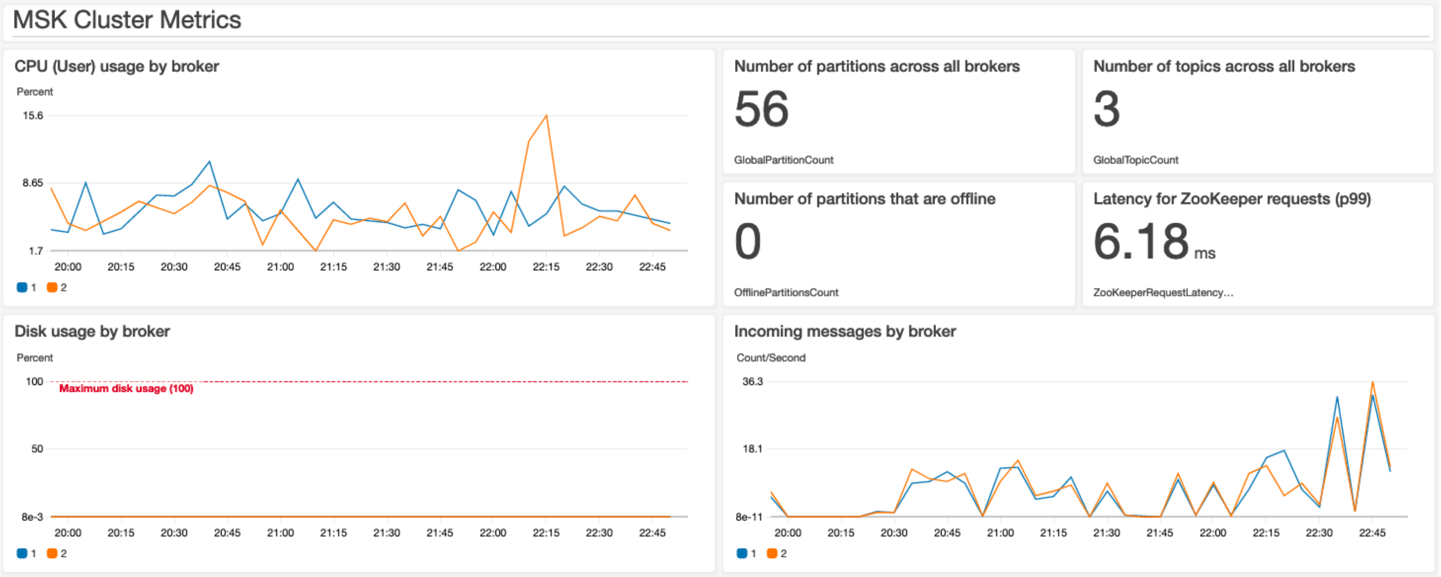
The solution utilizes managed and serverless services, to minimize the environmental impact of the backend services. The solution Serverless design (using Lambda, SQS, API Gateway, and S3) and the use of managed services (such as Kinesis Data Streams) are aimed at reducing carbon footprint compared to the footprint of continually operating on-premises servers.

CloudWatch dashboards deployed by solution options

Option 1: Amazon MSK

CloudWatch dashboards and alerts

Option 1 deploys an Amazon CloudWatch dashboard that monitors the health of the Amazon MSK cluster. You can customize the dashboards and alerts using Amazon CloudWatch or the source code from the solution's GitHub repository.



MSK Cluster Metrics

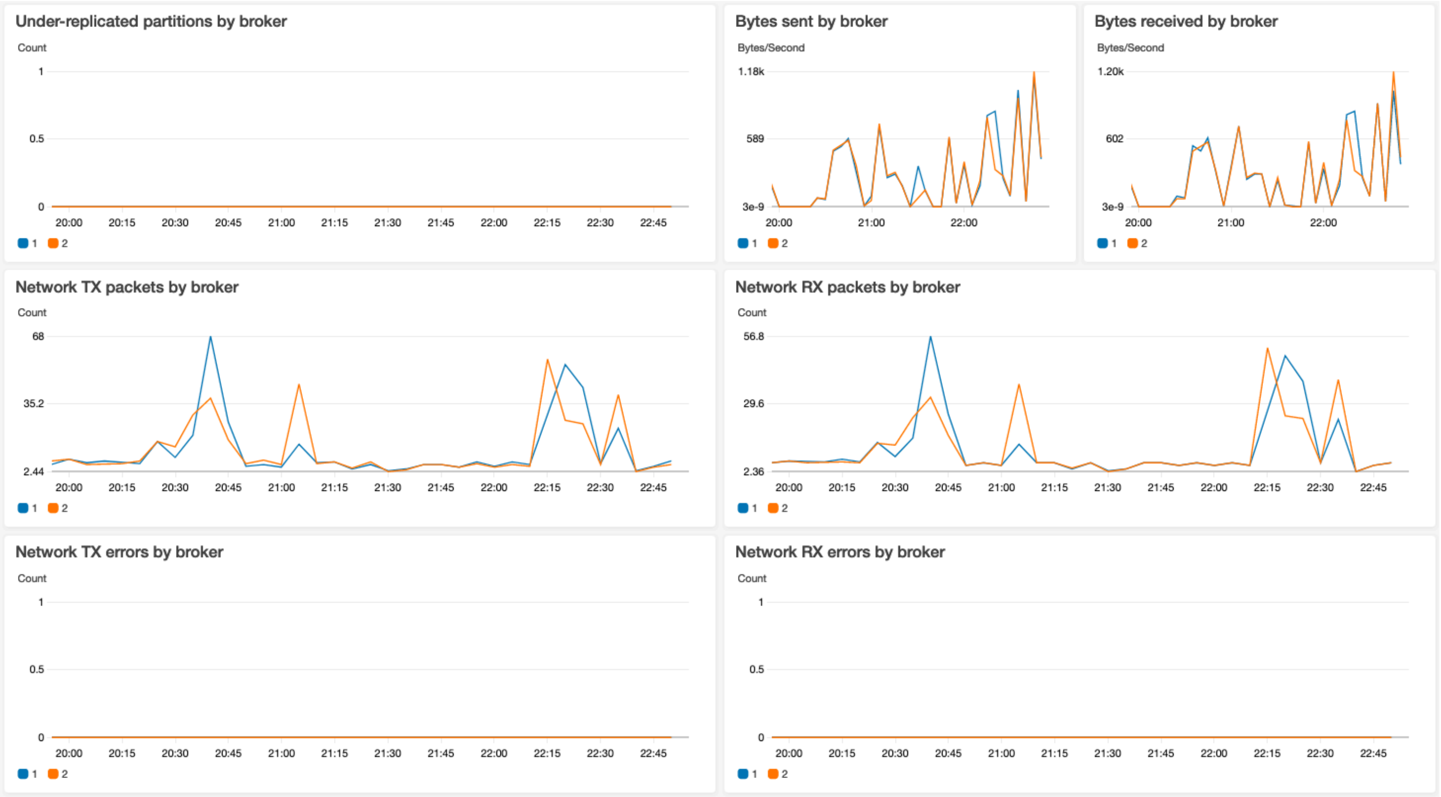
Number of partitions across all brokers: 56 (GlobalPartitionCount)

Number of topics across all brokers: 3 (GlobalTopicCount)

Number of partitions that are offline: 0 (OfflinePartitionsCount)

Latency for ZooKeeper requests (p99): 6.18 ms (ZooKeeperRequestLatency...)

Amazon MSK health metrics on the CloudWatch dashboard (lower)



Amazon MSK health metrics on the CloudWatch dashboard (lower)

Option 2: Amazon MSK with AWS Lambda

This option does not deploy a CloudWatch dashboard.

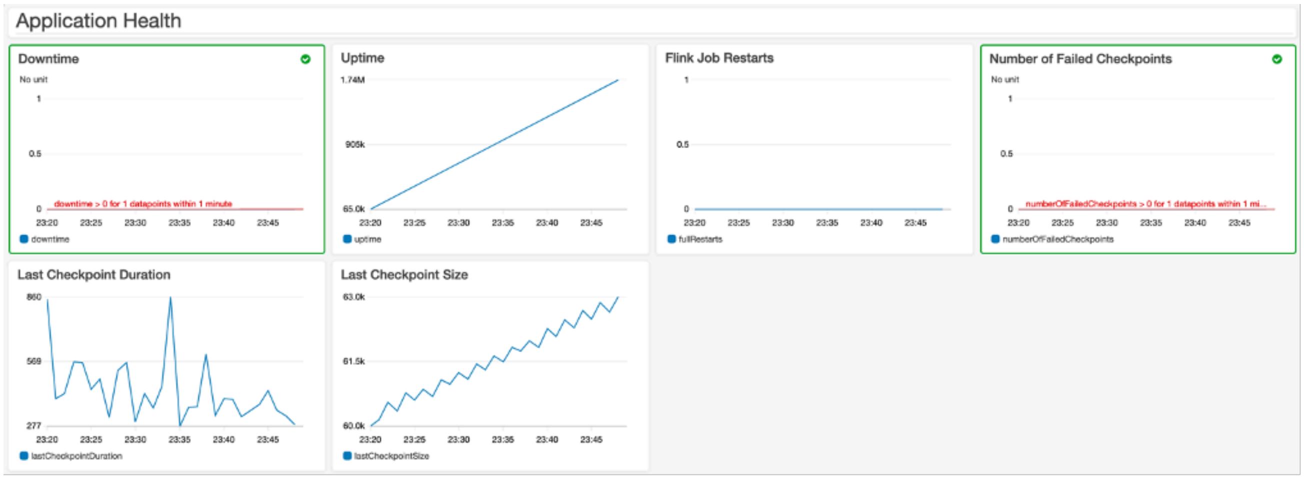
Option 3: Amazon MSK with AWS Lambda and Amazon Kinesis Firehose

This option does not deploy a CloudWatch dashboard.

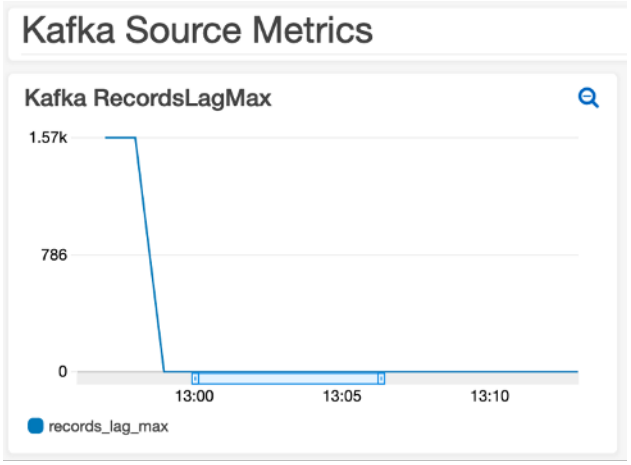
Option 4: Amazon MSK, Amazon Managed Service for Apache Flink, and Amazon S3

CloudWatch dashboards and alerts

Option 4 deploys an Amazon CloudWatch dashboard that monitors the health of the Apache Flink application. You can customize the dashboards and alerts using either Amazon CloudWatch, or the source code from the solution’s GitHub repository.



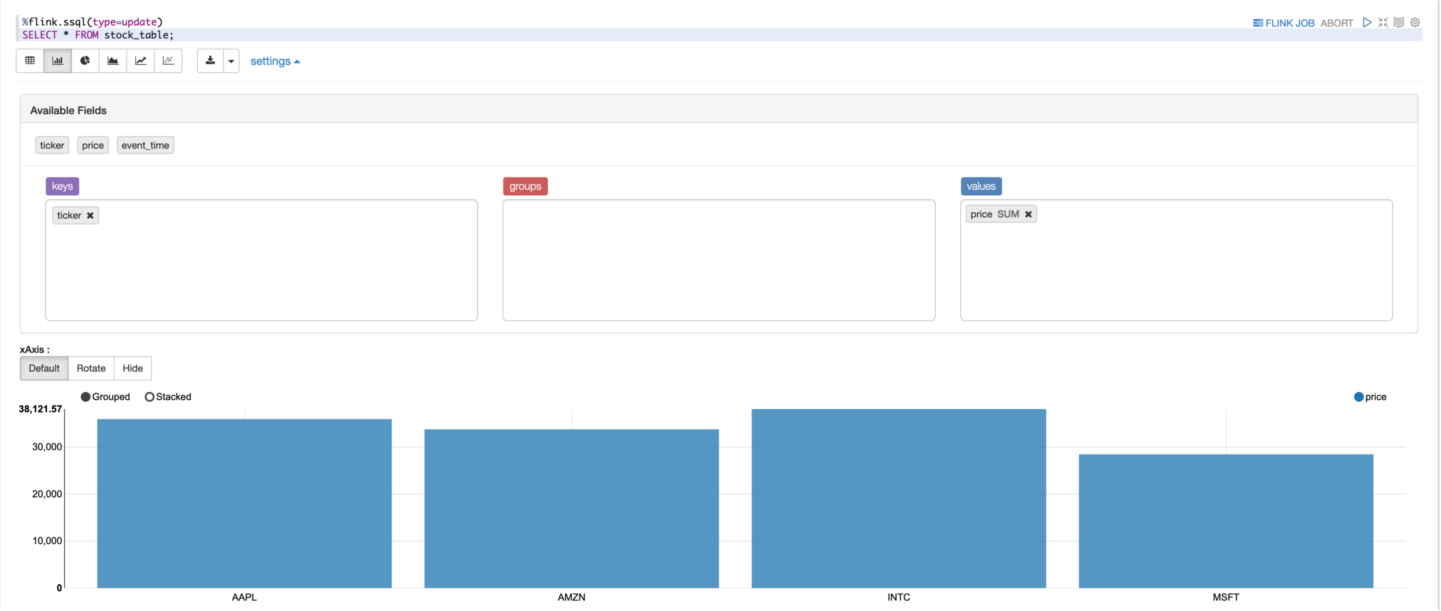
Application Health on the CloudWatch dashboard



Kafka Source Metrics on the CloudWatch dashboard

Studio notebook

Option 4 deploys an Amazon Managed Service for Apache Flink Studio notebook powered by Apache Zeppelin and Apache Flink to interactively analyze streaming data.



```
%flink.ssql(type=update)
SELECT * FROM stock_table;
```

Available Fields

ticker price event_time

keys: ticker

groups:

values: price SUM

xAxis : Default Rotate Hide

● Grouped ○ Stacked

Example query on the Studio notebook

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about AWS security, refer to AWS Cloud Security.

IAM roles

AWS Identity and Access Management (IAM) roles enable customers to assign granular access policies and permissions to services and users in the AWS Cloud. This solution creates IAM roles for communication between services. For more information, refer to Providing Access to an AWS Service in the *IAM User Guide*.

Security groups

This solution creates a security group for the Amazon MSK cluster so that it can communicate with the other solution components. This security group only includes the minimal rules required for Apache Kafka to work properly.

Auditing

Each AWS service included in this solution is integrated with AWS CloudTrail, which captures all API calls. For more details, refer to the following documentation:

- Logging Amazon MSK API Calls with AWS CloudTrail
- Logging AWS Lambda API calls with AWS CloudTrail
- Logging Managed Service for Apache Flink API Calls with AWS CloudTrail

AWS CloudFormation templates

This solution uses AWS CloudFormation to automate the deployment of the Streaming Data Solution for Amazon Amazon MSK in the AWS Cloud. You can download the following CloudFormation templates to deploy and customize to meet your needs:

View template

Option 1: **streaming-data-solution-for-msk.template** - Use this template to launch this solution using Amazon MSK.

View template

Option 2: **streaming-data-solution-for-msk-using-aws-lambda.template** - Use this template to launch this solution using Amazon Managed Streaming for Apache Kafka (Amazon MSK) and AWS Lambda.

View template

Option 3: **streaming-data-solution-for-msk-using-aws-lambda-and-data-firehose.template** - Use this template to launch the solution using Amazon MSK, Lambda, and Amazon Data Firehose.

View template

Option 4: **streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3.template** - Use this template to launch this solution using Amazon MSK, Amazon Managed Service for Apache Flink, and Amazon S3.

Deploy the solution

Prerequisites

Choose one of the following AWS CloudFormation templates to deploy, then follow the step-by-step instructions for your selected template:

- **Option 1:** Deploy the `streaming-data-solution-for-msk.template` AWS CloudFormation template using Amazon Managed Streaming for Apache Kafka (Amazon MSK).
- **Option 2:** Deploy the `streaming-data-solution-for-msk-using-aws-lambda.template` AWS CloudFormation template using Amazon MSK and AWS Lambda.
- **Option 3:** Deploy the `streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose.template` AWS CloudFormation template using Amazon MSK, Lambda, and Amazon Data Firehose.
- **Option 4:** Deploy the `streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3.template` AWS CloudFormation template using Amazon MSK, Amazon Managed Service for Apache Flink, and Amazon S3.

Option 1: Deploy the streaming-data-solution-for-msk CloudFormation template

Before you launch this template, review the architecture and other considerations in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 25-30 minutes

Deployment overview

Use the following steps to deploy this solution on AWS. For detailed instructions, follow the links for each step.

Step 1. Launch the stack

1. Launch the AWS CloudFormation template into your AWS account.

2. Review the template parameters, and adjust if necessary.

Step 2. (Optional) Create a topic that produces and consumes data

Step 1. Launch the stack

> **Note**
>
> You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost section for more details. For full details, refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and use the button below to launch the `streaming-data-solution-for-msk.template` AWS CloudFormation template.

   Launch solution

   Alternatively, you can download the template as a starting point for your own implementation.

2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar.

   > **Note**
   >
   > This template uses Amazon MSK, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where Amazon MSK is available. For the most current availability by Region, refer to the AWS Regional Services List.

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.

4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS Limits in the *AWS Identity and Access Management User Guide*.

5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Broker configuration		
Apache Kafka version **(KafkaVersion)**	`2.8.1`	Apache Kafka version on the brokers.
Number of broker nodes **(NumberBrokerNodes)**	`3`	Number of broker nodes you want in the cluster (must be a multiple of the number of subnets).
Broker instance type **(BrokerInstanceType)**	`kafka.m5.large`	Amazon EC2 instance type that Amazon MSK uses when it creates your brokers.
Monitoring level **(MonitoringLevel)**	`DEFAULT`	Level of monitoring for the cluster. The available options include `DEFAULT`, `PER_BROKER` , `PER_TOPIC_PER_BROKER` and `PER_TOPIC_PER_PARTITION` .
Amazon EBS storage volume per broker (in GiB) (EbsVolumeSize)	`1000`	Size (in GiB) of the storage volume in each broker node. The allowed range is from 1 to 16384.
Access control configuration		

Parameter	Default	Description
Method Amazon MSK uses to authenticate clients **(AccessControlMethod)**	`IAM role-based authentication`	The available options are `Unauthenticated access`, `IAM role-based authentication` , and `SASL/SCRAM authentication` .
Networking configuration		
Cluster VPC **(BrokerVpcId)**	*`<Requires input>`*	VPC where the cluster launch.
Cluster subnets **(BrokerSubnetIds)**	*`<Requires input>`*	List of subnets in which brokers are distributed (must contain between 2 and 3 items).
Client configuration		
Instance type **(ClientInstanceType)**	`t3.small`	Instance type for the client instance.
Amazon Machine Image **(ClientAmiId)**	`1`	Amazon Machine Image (AMI) ID for the client instance.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 25 minutes.

Note

This solution includes the `solution-helper` Lambda function, which runs only during initial configuration. This function is only created if you start the collection of operational metrics. For details, refer to Anonymized data collection.

Step 2. (Optional) Create a topic that produces and consumes data

After the stack is created, you can use the Amazon EC2 client instance to interact with the Amazon MSK cluster.

1. Sign in to the Amazon MSK console and, from the left menu pane, select **Clusters**.
2. On the **Amazon MSK** page, select `kafka-cluster-`*`<account-id>`*.
3. Choose **View client information** then copy the values for **ZooKeeper connection** and **Bootstrap servers**.
4. Navigate to the AWS Systems Manager console and, from the left menu pane under **Instances and Nodes**, select **Session Manager**.
5. On the **AWS Systems Manager** page, choose **Start session**.
6. On the **Start a session** page, select the *`<KafkaClient>`* and choose **Start session**.

   Refer to the AWS CloudFormation **Outputs** tab for the Amazon EC2 instance ID.
7. In the console window, run the following command to create a topic:

```
    sudo su
    cd /home/kafka/bin
    ./kafka-topics.sh --create --zookeeper <ZookeeperConnectString> --replication-factor 2 --partitions 2 --topic msk-serverless-tutorial/home/kafka/bin
./kafka-topics.sh --create --zookeeper<zookeeper-connection-string> --replication-factor 2 --partitions 2 --topic MyTopic
./kafka-console-producer.sh --broker-list<broker-list> --producer.config config-file --topic MyTopic
```

Note

The client configuration file depends on the access control method selected when launching the stack. For **Unauthenticated access**, use `client-ssl.properties`; for **IAM**

role-based authentication, use `client-iam.properties`; and for **SASL/SCRAM**, use `client-sasl.properties`

Option 2: Deploy the streaming-data-solution-for-msk-using-aws-lambda CloudFormation template

Before you launch this template, review the architecture and other considerations in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately five minutes

Launch the Stack

> **Note**
>
> You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost section for more details. For full details, refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and use the button below to launch the `streaming-data-solution-for-msk-using-aws-lambda` AWS CloudFormation template.

   Launch solution

   Alternatively, you can download the template as a starting point for your own implementation.
2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS Limits in the *AWS Identity and Access Management User Guide*.

5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
AWS Lambda consumer configuration		
ARN of the MSK cluster **(ClusterArn)**	*<Requires input>*	ARN of the Amazon MSK cluster.
Maximum number of items to retrieve in a single batch **(BatchSize)**	`100`	The maximum number of records to retrieve in a single batch. The allowed range is from 1 to `10000`.
Name of a Kafka topic to consume **(TopicName)**	*<Requires input>*	The name of the Apache Kafka topic to consume.
Secret ARN for SASL/SCRA M authentication **(SecretArn)**	*<Optional input>*	ARN of the AWS Secrets Manager secret containing the sign-in credentials to be used for authentication with the cluster.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately five minutes.

Option 3: Deploy the streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose CloudFormation template

Before you launch this template, review the architecture and other considerations in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 10 minutes

Launch the stack

> **Note**
>
> You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost section for more details. For full details, refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and use the button below to launch the `streaming-data-solution-for-msk-using-aws-lambda-and-kinesis-data-firehose` AWS CloudFormation template.

   Launch solution

   Alternatively, you can download the template as a starting point for your own implementation.
2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS Limits in the *AWS Identity and Access Management User Guide*.

5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
AWS Lambda consumer configuration		
ARN of the MSK cluster **(ClusterArn)**	*<Requires input>*	ARN of the Amazon MSK cluster.
Maximum number of items to retrieve in a single batch **(BatchSize)**	`100`	The maximum number of records to retrieve in a single batch. The allowed range is from 1 to `10000` hours.
Name of a Kafka topic to consume **(TopicName)**	*<Requires input>*	The name of the Apache Kafka topic to consume.
Secret ARN for SASL/SCRAM authentication (SecretArn)	*<Optional input>*	ARN of the AWS Secrets Manager secret containing the sign-in credentials to be used for authentication with the cluster.
Amazon Data Firehose configuration		
Size of the buffer (in MBs) that incoming data is buffered before delivery **(BufferingSize)**	5	The size to buffer incoming data before delivering to S3. The allowed range is from 1 to 128.

Parameter	Default	Description
Length of time (in seconds) that incoming data is buffered before delivery **(BufferingInterval)**	`300`	The amount of time to buffer incoming data before delivering to S3. The allowed range is from `60` to `900`.
Compression format for delivered data in Amazon S3 **(CompressionFormat)**	`GZIP`	The format of data once it's delivered to S3. Allowed values are `GZIP`, `HADOOP_SNAPPY` , `Snappy`, `UNCOMPRESSED` , and `ZIP`.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately ten minutes.

Option 4: Deploy the streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3 CloudFormation template

Before you launch this template, review the architecture and other considerations in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 10 minutes

Step 1. Launch the stack

Note

You are responsible for the cost of the AWS services used while running this solution. Refer to the Cost section for more details. For full details, refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and use the button below to launch the `streaming-data-solution-for-msk-using-kinesis-data-analytics-and-amazon-s3` AWS CloudFormation template.

   Launch solution

   Alternatively, you can download the template as a starting point for your own implementation.
2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS Limits in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Amazon MSK cluster configuration		
ARN of the MSK cluster **(ClusterArn)**	*<Requires input>*	ARN of the Amazon MSK cluster.
Amazon Managed Service for Apache Flink configuration		

Parameter	Default	Description
Monitoring log level (LogLevel)	`INFO`	The level of detail of the CloudWatch logs for an application. The available options include `DEBUG`, `ERROR`, `INFO`, and `WARN`. For information about choosing a log level, refer to Application Monitoring Levels in the *Amazon Kinesis Data Analytics Developer Guide*.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately ten minutes.

Step 2. Post-configuration steps

By default, the Studio notebook will not run after the stacks are created. Use the following process to start the Studio notebook.

1. Sign in to the Amazon Kinesis console and, from the left menu pane, select **Analytics applications**.
2. On the **Amazon Managed Service for Apache Flink** page, go to the **Studio** tab, and select **Kda*<studio-notebook-name>***.
3. Choose **Run**.

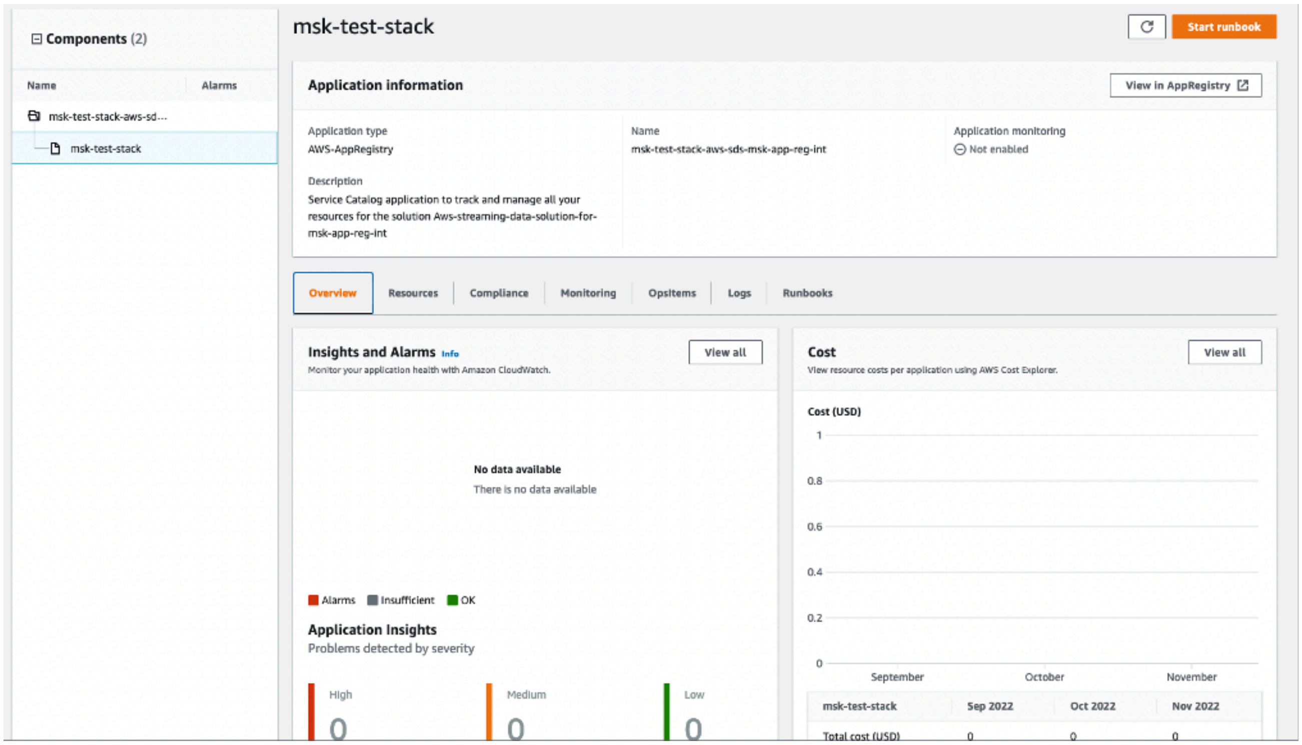
Monitoring this solution with AWS Service Catalog AppRegistry

The Streaming Data Solution for Amazon MSK solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both AWS Service Catalog AppRegistry and AWS Systems Manager Application Manager.

AWS Systems Manager Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the resources of this solution in the context of an application, such as deployment status, CloudWatch alarms, resource configurations, and operational issues.

The following figure depicts an example of the application view for the Streaming Data Solution for Amazon MSK stack in Application Manager.



msk-test-stack

Application information

Application type
AWS-AppRegistry

Name
msk-test-stack-aws-sds-msk-app-reg-int

Application monitoring
⊖ Not enabled

Description
Service Catalog application to track and manage all your resources for the solution Aws-streaming-data-solution-for-msk-app-reg-int

Insights and Alarms
Monitor your application health with Amazon CloudWatch.

No data available
There is no data available

Application Insights
Problems detected by severity

High 0
Medium 0
Low 0

Cost
View resource costs per application using AWS Cost Explorer.

Cost (USD)

msk-test-stack	Sep 2022	Oct 2022	Nov 2022
Total cost (USD)	0	0	0

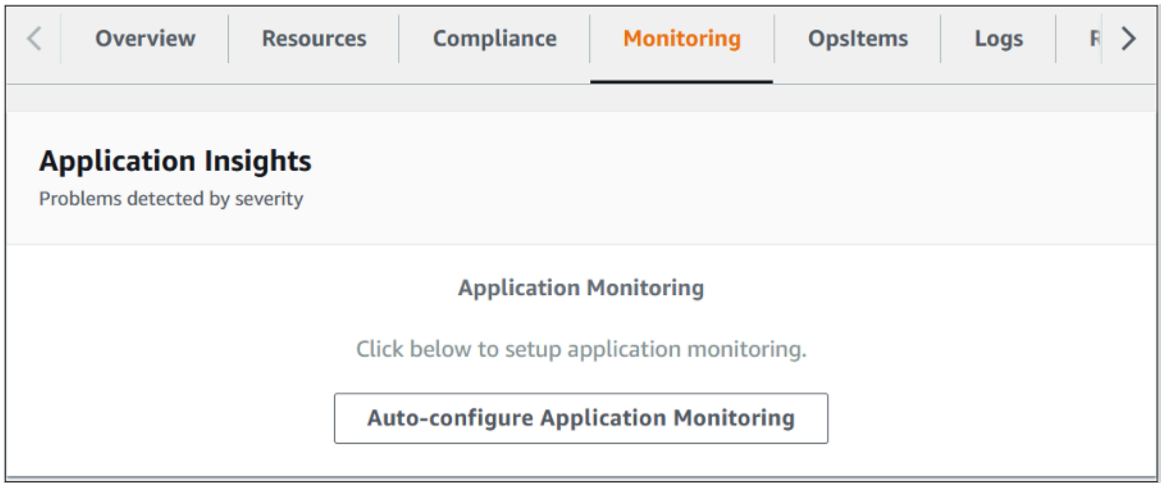
Streaming Data Solution for Amazon MSK Application Manager

> **Note**
>
> AWS Cost Explorer must be activated. It is not activated by default.

Activate CloudWatch Application Insights

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose **AppRegistry applications**.
4. In **AppRegistry applications**, search for the application name for this solution and select it.

   The next time you open Application Manager, you can find the new application for your solution in the **AppRegistry application** category.
5. In the **Components** tree, choose the application stack you want to activate.
6. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Monitoring**.



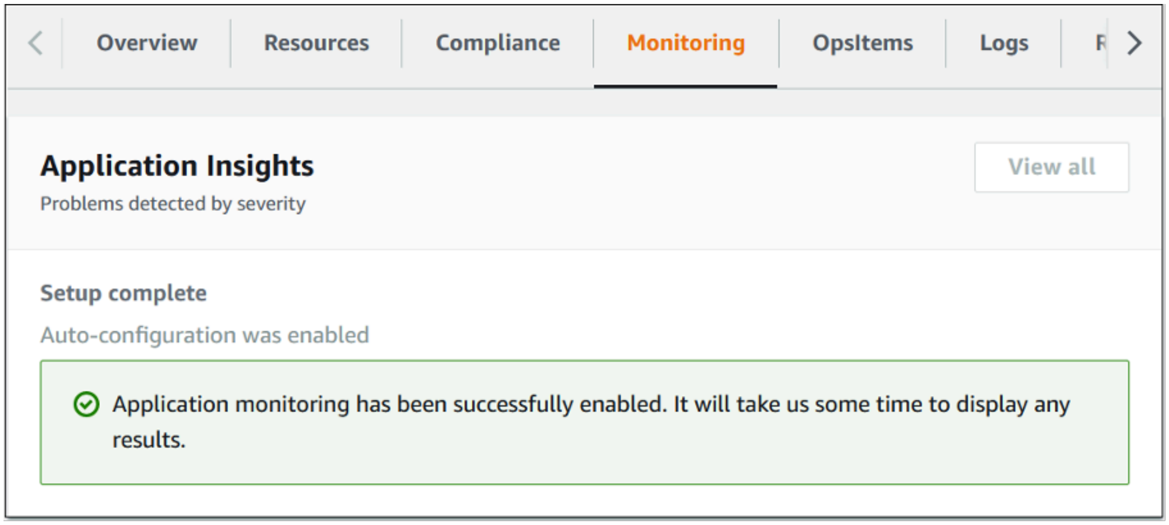
Application Insights

Problems detected by severity

Application Monitoring

Click below to setup application monitoring.

Monitoring for your applications is now activated and the following status box appears:



Application Insights

Problems detected by severity

Setup complete

Auto-configuration was enabled

Application monitoring has been successfully enabled. It will take us some time to display any results.

Activate AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer which must be first activated. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time. To activate Cost Explorer for the solution:

1. Sign in to the AWS Cost Management console.
2. In the navigation pane, select **Cost Explorer**.
3. On the **Welcome to Cost Explorer** page, choose **Launch Cost Explorer**.

The activation process can take up to 24 hours to complete. Once activated, you can open the Cost Explorer user interface to further analyze cost data for the solution.

Activate cost allocation tags associated with the solution

After you activate Cost Explorer, you must activate a cost allocation tag to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization. To activate cost allocation tags:

1. Sign in to the AWS Billing and Cost Management console.
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the Cost allocation tags page, filter for the `AppManagerCFNStackKey` tag, then select the tag from the results shown.

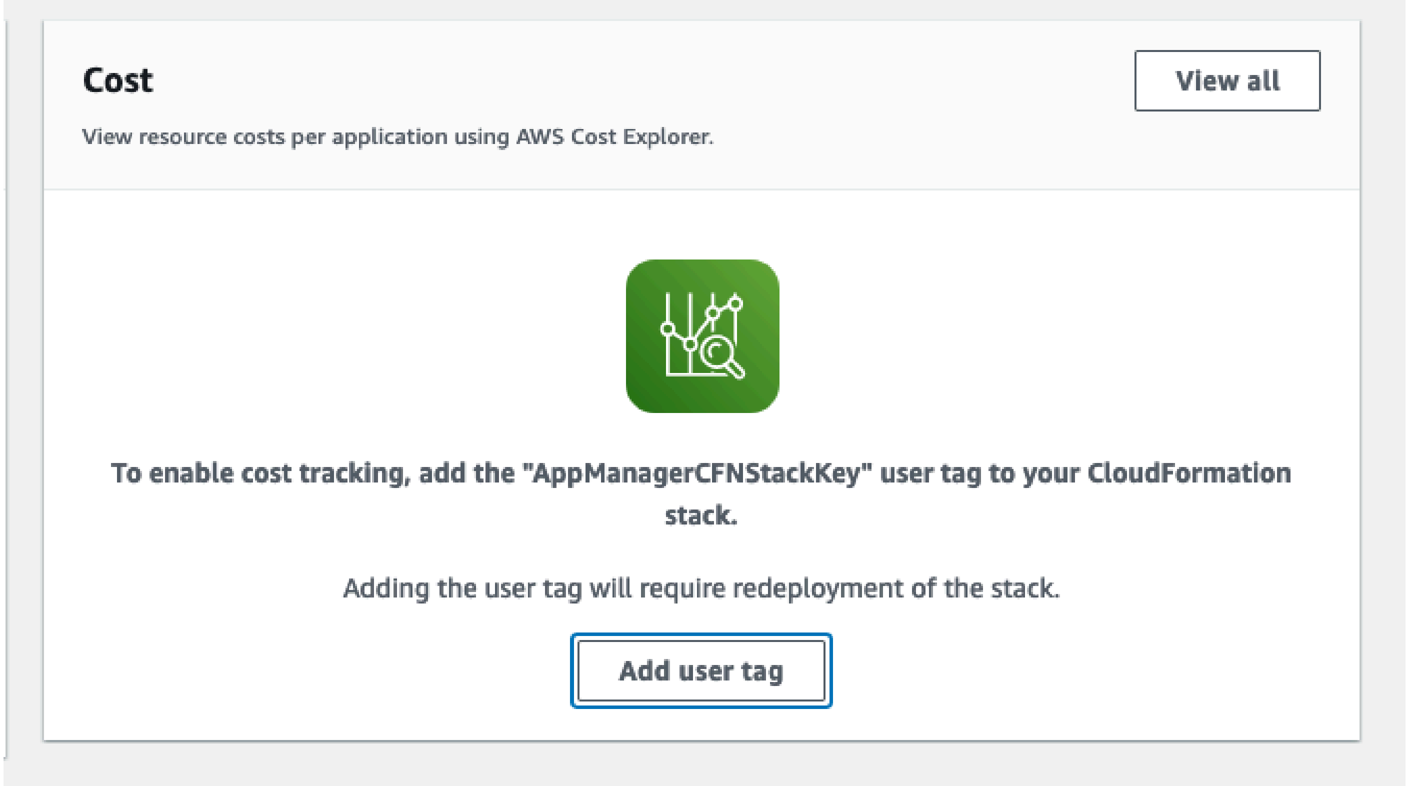
4. Choose **Activate**

The activation process can take up to 24 hours to complete and the tag data to appear.

Confirm cost tags associated with the solution

After you activate cost allocation tags associated with the solution, you must confirm the cost allocation tags to see the costs for this solution. To confirm cost allocation tags:

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose the application name for this solution and select it.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.



Cost

View resource costs per application using AWS Cost Explorer.

To enable cost tracking, add the "AppManagerCFNStackKey" user tag to your CloudFormation stack.

Adding the user tag will require redeployment of the stack.

5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Additional resources

AWS services

- Amazon CloudWatch
- Amazon Elastic Compute Cloud
- Amazon Data Firehose
- Amazon Managed Streaming for Apache Kafka
- Amazon Simple Storage Service
- AWS CloudFormation
- AWS Identity and Access Management
- AWS Lambda
- AWS Systems Manager
- Amazon Managed Service for Apache Flink

AWS documentation

Best practices for monitoring and data protection:

- Security in Amazon Managed Streaming for Apache Kafka
- Using Lambda with Amazon MSK
- Controlling Access to Apache ZooKeeper
- Security in Amazon Managed Service for Apache Flink
- Viewing Managed Service for Apache Flink Metrics and Dimensions

Amazon MSK Labs

- The Amazon MSK Labs are a learning resource that take you through getting started, a use case example of ingesting and analyzing real-time clickstream data, and best practices for migrating your self-managed Apache Kafka cluster to Amazon MSK. They also showcase how to leverage advanced Amazon MSK features (such Cruise Control, TLS mutual authentication, and open monitoring), which can be applied to clusters created using the solution.

Uninstall the solution

You can uninstall the Streaming Data Solution for Amazon MSK using the AWS Management Console or the AWS Command Line Interface (AWS CLI). The CloudWatch dashboard (along with any changes made directly to CloudWatch) is deleted with the solution stack. However, the Amazon Simple Storage Service (Amazon S3) bucket and Amazon CloudWatch Logs created by this solution must be manually deleted.

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. On the **Stacks** page, select the solution stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to What Is the AWS Command Line Interface in the *AWS CLI User Guide*. After confirming the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <cloudformation-stack-name>
```

Replace *<cloudformation-stack-name>* with the name of your CloudFormation stack.

Deleting the Amazon S3 buckets

To prevent against accidental data loss, this solution is configured to retain the Amazon S3 buckets if you choose to delete the AWS CloudFormation stack. After uninstalling the solution, you can manually delete the S3 buckets if you do not need to retain the data. Use the following procedure to delete the Amazon S3 buckets.

1. Sign in to the Amazon S3 console.
2. Choose **Buckets** from the left navigation pane.
3. Locate the *<stack-name>* S3 buckets.
4. Select one of the S3 buckets and choose **Delete**.

Repeat the steps until you have deleted all the `<stack-name>` S3 buckets.

Alternatively, you can configure the AWS CloudFormation template to delete the Amazon S3 buckets automatically. Before deleting the stack, change the deletion behavior in the AWS CloudFormation DeletionPolicy attribute.

Deleting the CloudWatch Logs

This solution retains the CloudWatch Logs if you decide to delete the AWS CloudFormation stack to prevent against accidental data loss. After uninstalling the solution, you can manually delete the logs if you do not need to retain the data. Use the following procedure to delete the CloudWatch Logs.

1. Sign in to the Amazon CloudWatch console.
2. Choose **Log Groups** from the left navigation pane.
3. Locate the log groups created by the solution.
4. Select one of the log groups.
5. Choose **Actions** and then choose **Delete**.

Repeat the steps until you have deleted all the solution log groups.

Developer guide

This section provides the source code for this solution.

Source code

Visit our GitHub repository to download the source files for this solution and to share your customizations with others.

The AWS Cloud Development Kit (AWS CDK) generates the Streaming Data Solution for Amazon Kinesis templates. See the README.md file for additional information.

Reference

This section includes information about an optional feature for collecting unique metrics for this solution and a list of builders who contributed to this solution.

Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When enabled, the following information is collected and sent to AWS:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each solution deployment
- **Timestamp** - The UTC formatted timestamp of when the event occurred
- **Data** - The Region where the stack launched, request type (whether the stack was created, updated, or deleted), and details about the option chosen (for example, shard count, whether enhanced monitoring was enabled, buffering size, etc.). For example:

```
{'Pattern': 'KdsKdfS3', 'RetentionHours': '24',
'CompressionFormat': 'GZIP', 'BufferingInterval': '300',
'ShardCount': '2', 'EnhancedMonitoring': 'false', 'Version': 'v1.2.0',
'BufferingSize': '5', 'Region': 'us-east-1', 'RequestType': 'Create'}
```

Note that AWS owns the data gathered through this survey. Data collection is subject to the AWS Privacy Policy. To opt out of this feature, modify the AWS CloudFormation template mapping section:

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
"Send" : {
"AnonymousUsage" : { "Data" : "Yes" }
},
```

to:

```
"Send" : {
"AnonymousUsage" : { "Data" : "No" }
},
```

4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, **Specify template** section, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Launch the stack in the Automated deployment section of this guide.

Contributors

- Mukit Bin Momin
- Tarek Abdunabi
- Daniel Pinheiro
- Morris Estepa
- Abhay Joshi

Revisions

Date	Change
November 2020	Initial release
January 2021	Release v1.3.0: Added support for Apache Kafka 2.7.0; added pattern for integration between Amazon MSK and Amazon Managed Service for Apache Flink. For more information, refer to the CHANGELOG.md file in the GitHub repository.
April 2021	Release v1.4.0: Added new parameter that specifies the size for the storage in each of the broker nodes; Added support for partition-level monitoring. For more information, refer to the CHANGELOG.md file in the GitHub repository.
May 2021	Release v1.4.1: Added Support for Apache Kafka versions 2.8.0 and 2.6.2. For more information, refer to the CHANGELOG.md file in the GitHub repository.
July 2021	Release v1.5.0: Added support for IAM access control and SASL/SCRAM authentication; Added support for Apache Kafka version 2.7.1; Fixed location of GitHub repository for MSK Labs assets. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2021	Release v1.6.0: Added support for clusters secured by IAM Access Control in options 2 and 3; Updated option 4 to use Amazon Managed Service for Apache Flink Studio, which offers a serverless notebook to perform live data exploration. For more information, refer to the CHANGELOG.md file in the GitHub repository.
July 2022	Release v1.6.1: Security updates for the Gson package and the minimist and vm2 npm packages. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Date	Change
September 2022	Release v1.6.2: Security patch for vm2 npm package. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2022	Release v1.7.0: Npm security patches. Application Registry integration. For more information, refer to the CHANGELOG.md file in GitHub repository for details.
December 2022	Release v1.7.1: Removed "AWS" prefix from Service Catalog AppRegistry and Attribute Group Name to correct a common error. For more information, refer to the CHANGELOG.md file in GitHub repository for details.
January 2023	Release v1.7.2: Security patch. For more information, refer to the CHANGELOG.md file in GitHub repository for details.
April 2023	Release v1.7.3: Security patching for npm packages. For more information, refer to the CHANGELOG.md file in GitHub repository for details.
April 2023	Release v1.7.4: Mitigated impact caused by new default settings for S3 Object Ownership (ACLs disabled) for all new S3 buckets. Security patching for maven packages. For more information, refer to the CHANGELOG.md file in GitHub repository for details.
April 2023	Release v1.7.5: Security patching for npm packages. For more information, refer to the CHANGELOG.md file in GitHub repository for details.
May 2023	Release v1.7.6: Minor updates and bug fixes. For more information, refer to the CHANGELOG.md file in GitHub repository for details.

Date	Change
June 2023	Release v1.7.7: Security patching for npm packages. Update logical ID of AppRegistry resources to prevent CloudForm ation stack update failures. For more information, refer to the CHANGELOG.md file in GitHub repository for details.
July 2023	Release v1.7.8: Security patching for Python packages. For more information, refer to the CHANGELOG.md file in GitHub repository for details.
September 2023	Release v1.8.0: Migrate to new SDKs. Upgrade Lambda runtimes. Security patches. For more information, refer to the CHANGELOG.md file in the GitHub repository.
October 2023	Release v1.8.1: Security patch. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2023	Documentation update: Added Confirm cost tags associate d with the solution to the Monitoring the solution with AWS Service Catalog AppRegistry section.
February 2024	Release v1.9.0: Upgrade Apache Flink and MSK. Encrypt data at-rest in Glue Data Catalog. Upgrade Lambda runtimes. Security patches. For more information, refer to the CHANGELOG.md file in the GitHub repository.
June 2024	Release v1.9.1: Onboarded to CloudFormation Guard scanning. Upgraded and patched dependencies. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Streaming Data Solution for Amazon MSK is licensed under the terms of the of the Apache License Version 2.0 available at The Apache Software Foundation.